Randomness and Degree Theory for Infinite Time Register Machines

Merlin Carl

Abstract

A concept of randomness for infinite time register machines (ITRMs) is defined and studied. In particular, we show that for this notion of randomness, computability from mutually random reals implies computability and that an analogue of van Lambalgen's theorem holds. This is then applied to obtain results on the structure of ITRM-degrees. Finally, we consider autoreducibility for ITRMs and show that randomness implies non-autoreducibility. This is an expanded and amended version of [4].

1 Introduction

Martin-Löf-randomness (ML-randomness, see e.g. [6]) provides an intuitive and conceptually stable clarification of the informal notion of a random sequence over a finite alphabet. The guiding idea of ML-randomness is that a sequence of 0 and 1 is random if and only if it has no special properties, where a special property should be a small (e.g. measure 0) set of reals that is in some way accessible to a Turing machine.

Since its introduction, several variants of this general approach to defining randomness have been considered; a recent example is the work of Hjorth and Nies on Π^1_1-randomness and a Π^1_1 version of ML-randomness, which led to interesting connections with descriptive set theory ([8]).

We are interested in obtaining a similar notion based on machine models of transfinite computations. In this paper, we will exemplarily consider infinite time register machines. Infinite Time Register Machines (ITRMs), introduced in [9] and further studied in [10], work similar to the classical unlimited register machines (URMs) described in [5]. In particular, they use finitely many registers each of which can store a single natural number. The difference is that ITRMs use transfinite ordinal running time: The state of an ITRM at a successor ordinal is obtained as for URMs. At limit times, the

program line is the inferior limit of the earlier program lines and there is a similar limit rule for the register contents. If the inferior limit of the earlier register contents is infinite, the register is reset to 0.

Classical Turing machines, due to the finiteness of their running time, have the handicap that the only decidable null set of reals is the empty set: If a real x is accepted by a classical Turing machine M within n steps, then M will also accept every y agreeing with x on the first n bits. In the definition of ML-randomness, this difficulty is overcome by merely demanding the set X in question to be effectively approximated by a recursively enumerable sequence of sets of intervals with controlled convergence behaviour. For models of transfinite computations, this trick is unnecessary: The decidable sets of reals form a rich class (including all ML-tests and, by [9], all Π^1_1-sets). This is still a plausible notion of randomness, since elements of an ITRM-decidable meager set can still be reasonably said to have a special property. In fact, some quite natural properties like coding a well-ordering can be treated very conveniently with our approach. Hence, we define:

Definition 1. $X \subseteq \mathfrak{P}(\omega)$ is called ITRM-decidable if and only if there is an ITRM-program P such that $P^x \downarrow= 1$ if and only if $x \in X$ and $P^x \downarrow= 0$, otherwise. In this case we say that P decides X. P is called deciding if and only if there is some X such that P decides X. We say that X is decided by P in the oracle y if and only if $X = \{x \mid P^{x\oplus y} \downarrow= 1\}$ and $\mathfrak{P}(\omega) - X = \{x \mid P^{x\oplus y} \downarrow= 0\}$. In this case, we also say that P^y decides X. The other notions relativize in the obvious way.

Definition 2. Recall that a set $X \subseteq \mathfrak{P}(\omega)$ is meager if and only if it is a countable union of nowhere dense sets. $X \subseteq \mathfrak{P}(\omega)$ is an ITRM-test if and only if X is ITRM-decidable and meager. $x \subseteq \omega$ is ITRM-c-random (where c stands for 'category') if and only if there is no ITRM-test X such that $x \in X$. $x \subseteq \omega$ is ITRM-random if and only if there is no ITRM-decidable set $X \ni x$ of Lebesgue measure 0. If there is no such set decidable in the oracle $y \subseteq \omega$, then x is called ITRM-c-random/ITRM-random relative to y.

Remark: Expect for the section on autoreducibility, we will mostly be concerned with ITRM-c-randomness. This obviously deviates from other definitions of randomness in that we use meager sets rather than null sets as our underlying notion of 'small'. The reason is simply that this variant turned out to be much more convenient to handle for technical reasons. The use of category rather than measure gives this definition a closer resemblance to what is, in the classical setting, refered to as genericity (see e.g. section 2.24 of [6]). We still decided to use the term 'ITRM-c-randomness' to avoid confusion with the frequently used concept of Cohen genericity, hence reserving

the term 'ITRM-random' for reals that do not lie in any ITRM-decidable null set. We are pursuing the notion of ITRM-randomness in ongoing work. In contrast to strong Π^1_1-randomness ([8], [17]), it will be shown below that there is no universal ITRM-test.

We will now summarize some key notions and results on ITRMs that will be used in the paper.

Definition 3. For P a program, $x, y \in \mathfrak{P}(\omega)$, $P^x \downarrow = y$ means that the program P, when run with oracle x, halts on every input $i \in \omega$ and outputs 1 if and only if $i \in y$ and 0, otherwise. $x \subseteq \omega$ is ITRM-computable in the oracle $y \subseteq \omega$ if and only if there is an ITRM-program P such that $P^y \downarrow = x$, in which case we occasionally write $x \leq_{\text{ITRM}} y$. If y can be taken to be $\emptyset$, x is ITRM-computable. We denote the set of ITRM-computable reals by COMP.

Remark: We occasionally drop the ITRM-prefix as notions like 'computable' always refer to ITRMs in this paper.

Theorem 4. Let $x, y \subseteq \omega$. Then x is ITRM-computable in the oracle y if and only if $x \in L_{\omega_\omega^{\text{CK},y}}[y]$, where $\omega_i^{\text{CK},y}$ denotes the ith y-admissible ordinal.

Proof. This is a straightforward relativization of Theorem 5 of [15], due to P. Koepke. $\square$

Theorem 5. Let $\mathbb{P}_n$ denote the set of ITRM-programs using at most n registers, and let $(P_{i,n}|i \in \omega)$ enumerate $\mathbb{P}_n$ in some natural way. Then the bounded halting problem $H_n^x := \{i \in \omega | P_{i,n}^x \downarrow\}$ is computable uniformly in the oracle x by an ITRM-program (using more than n registers).

Furthermore, if $P \in \mathbb{P}_n$ and $P^x \downarrow$, then P^x halts in less than $\omega_{n+1}^{\text{CK},x}$ many steps. Consequently, if P is a halting ITRM-program, then P^x stops in less than $\omega_\omega^{\text{CK},x}$ many steps.

Proof. The corresponding results from [9] (Theorem 4) and [15] (Theorem 9) easily relativize. $\square$

Definition 6. For $x \subseteq \omega$, x'_{ITRM} denotes the set of $i \in \omega$ such that $P_i^x \downarrow$. x'_{ITRM} is called the ITRM-jump of x. We furthermore define the first ω many iterations of the ITRM-jump of x by $x^{(0)} = x$, $x_{\text{ITRM}}^{(i+1)} = (x_{\text{ITRM}}^{(i)})'$.

We will freely use the following standard propositions:

Proposition 7. Let $X \subseteq [0,1] \times [0,1]$ and $\tilde{X} := \{x \oplus y \mid (x, y) \in X\}$. Then X is meager/comeager/non-meager if and only if $\tilde{X}$ is meager/comeager/non-meager.

Proposition 8. If $X \subseteq [0, 1]$ is meager, then so are $X \oplus [0, 1]$ and $[0, 1] \oplus X$.

Most of our notation is standard. By a real, we mean an element of ${}^{\omega}2$. $L_\alpha[x]$ denotes the αth level of Gödel's constructible hierarchy relativized to x. For $a, b \subseteq \omega$, $a \oplus b$ denotes $\{p(i, j) \mid i \in a \wedge j \in b\}$, where $p : \omega \times \omega \to \omega$ is Cantor's pairing function. During the paper, we will frequently code countable $\in$-structures by reals. The idea here is the following: Let X be transitive and countable, and let $f : \omega \leftrightarrow X$ be a bijection. Then $(X, \in)$ is coded by the real $r_{X,f} := \{p(i, j) | i \in \omega \wedge j \in \omega \wedge f(i) \in f(j)\} \subseteq \omega$, and it is easy to re-obtain $(X, \in)$ from $r_{X,f}$. In general, a real x is a code for the transitive $\in$-structure $(X, \in)$ if and only if there is a bijection $f : \omega \to X$ such that $x = r_{X,f}$. If $(X, \in)$ is constructible and X is transitive and countable in L, then L contains a $<_L$-minimal code for $(X, \in)$; this code will be called the canonical code for $(X, \in)$ and denoted by $\mathrm{cc}(X)$.

2 Computability from random oracles

In this section, we consider the question which reals can be computed by an ITRM with an ITRM-c-random oracle. We start by recalling the following theorem from [3]. The intuition behind it is that, given a certain non-ITRM-computable real x, one has no chance of computing x from some randomly chosen real y.

Theorem 9. Let x be a real, Y be a set of reals such that x is ITRM-computable from every $y \in Y$. Then, if Y has positive Lebesgue measure or is Borel and non-meager, x is ITRM-computable.

Corollary 10. Let x be ITRM-c-random. Then, for all $i \in \omega$, $\omega_i^{\mathrm{CK},x} = \omega_i^{\mathrm{CK}}$.

Proof. Lemma 46 of [3] shows that $\omega_i^{\mathrm{CK},x} = \omega_i^{\mathrm{CK}}$ for all $i \in \omega$ whenever x is Cohen-generic over $L_{\omega_\omega^{\mathrm{CK}}}$ (see e.g. [3] or [16]) and that the set of Cohen-generics over $L_{\omega_\omega^{\mathrm{CK}}}$ is comeager. Hence $\{x | \omega_i^{\mathrm{CK},x} > \omega_i^{\mathrm{CK}}\}$ is meager. For each program P, the set of reals x such that $\omega_i^{\mathrm{CK},x} > \omega_i^{\mathrm{CK}}$ and P^x computes a code for $\omega_i^{\mathrm{CK},x}$ is decidable using the techniques developed in [1] and [2]. (The idea is to uniformly in the oracle x compute a real c coding $L_{\omega_{i+1}^{\mathrm{CK},x}}[x]$ in which the natural numbers m and n coding ω_i^{CK} and $\omega_i^{\mathrm{CK},x}$ can be identified in the oracle x, and then to check - using a halting problem solver for P, see Theorem 5 - whether P^x computes a well-ordering of the same order type as the element of $L_{\omega_{i+1}^{\mathrm{CK},x}}[x]$ coded by n and finally whether the element coded by m is an element of that coded by n.) Hence, if x is ITRM-c-random, then there can be no ITRM-program P computing such a code in the oracle

x. But a code for $\omega_i^{\mathrm{CK},x}$ is ITRM-computable in the oracle x for every real x and every $i \in \omega$. Hence, we must have $\omega_i^{\mathrm{CK},x} = \omega_i^{\mathrm{CK}}$ for every $i \in \omega$, as desired. $\square$

A notable conceptual difference between ITRM-c-randomness and Martin-Löf-randomness is the absence of a universal test for the former:

Theorem 11. There is no universal test for ITRM-c-randomness, i.e. the union of all ITRM-decidable meager sets is not ITRM-decidable.

Proof. Assume otherwise. Let U be the union of all ITRM-decidable meager sets and let P be an ITRM-program such that P decides U. Clearly, U, as a countable union of meager sets, is meager. Let P have n registers. Let M be the set of reals x that are Cohen-generic over $L_{\omega_{n+1}^{\mathrm{CK}}}$, but not over $L_{\omega_{n+2}^{\mathrm{CK}}}$. As the set of reals which are Cohen-generic over $L_{\omega_{n+2}^{\mathrm{CK}}}$ is comeager, M is a subset of a meager set and hence meager. It is also easy to see that M is ITRM-decidable: To decide M, compute a code for $L_{\omega_{n+2}^{\mathrm{CK}}}$; then, given a real y in the oracle, search through the dense subsets of Cohen-forcing in $L_{\omega_{n+2}^{\mathrm{CK}}}$ to see whether x intersects all those contained in $L_{\omega_{n+1}^{\mathrm{CK}}}$, but fails to intersect at least one dense subset contained in $L_{\omega_{n+2}^{\mathrm{CK}}}$.

Now pick $x \in M$. As $x \in M$ and M is ITRM-decidable and meager, we have $x \in U$, so $P^x \downarrow= 1$. By genericity of x over $L_{\omega_{n+1}^{\mathrm{CK}}}$ and the forcing theorem, there is a condition p of Cohen-forcing such that $p \subseteq x$ and $p \Vdash P^x \downarrow= 1$. Consequently, we have $p \Vdash P^y \downarrow= 1$ for every $L_{\omega_{n+1}^{\mathrm{CK}}}$-generic real y extending p. By absoluteness of computations, it follows that $P^y \downarrow= 1$ for each such y. Since P decides U, each such y hence belongs to U. But the set of these y is comeager in some interval and hence not meager. Hence U has a non-meager subset and his thus not meager, a contradiction. $\square$

Lemma 12. Let $a \subseteq \omega$ and suppose that z is Cohen-generic over $L_{\omega_\omega^{\mathrm{CK},a}+1}[a]$. Then $a \leq_{\mathrm{ITRM}} z$ if and only if a is ITRM-computable. Consequently (as the set $C_a := \{z \subseteq \omega \mid z \text{ is Cohen-generic over } L_{\omega_\omega^{\mathrm{CK},a}+1}[a]\}$ is comeager), the set $S_a := \{z \subseteq \omega \mid a \leq_{\mathrm{ITRM}} z\}$ is meager whenever a is not ITRM-computable.

Proof. Assume that z is Cohen-generic over $L_{\omega_\omega^{\mathrm{CK},a}+1}[a]$ and $a \leq_{\mathrm{ITRM}} z$. By the forcing theorem for provident sets (see e.g. Lemma 32 of [3]), there is an ITRM-program P and a forcing condition p such that $p \Vdash P^{\dot{G}} \downarrow= \check{a}$, where $\dot{G}$ is the canonical name for the generic filter and $\check{a}$ is the canonical name of a. Now, let y and z be mutually Cohen-generic over $L_{\omega_\omega^{\mathrm{CK},a}+1}[a]$ both extending p. Again by the forcing theorem and by absoluteness of computations, we must have $P^x \downarrow= a = P^y \downarrow$, so $a \in L_{\omega_\omega^{\mathrm{CK},x}}[x] \cap L_{\omega_\omega^{\mathrm{CK},y}}[y]$. By Corollary 10, $\omega_\omega^{\mathrm{CK},x} = \omega_\omega^{\mathrm{CK},y} = \omega_\omega^{\mathrm{CK}}$. By Lemma 28 of [3], we have $L_\alpha[x] \cap L_\alpha[y] = L_\alpha$

whenever x and y are mutually Cohen-generic over L_α and α is provident (see [16]). Consequently, we have:

$$a \in L_{\omega_\omega^{\mathrm{CK},x}}[x] \cap L_{\omega_\omega^{\mathrm{CK},y}}[y] = L_{\omega_\omega^{\mathrm{CK}}}[x] \cap L_{\omega_\omega^{\mathrm{CK}}}[y] = L_{\omega_\omega^{\mathrm{CK}}},$$

so a is ITRM-computable.

The comeagerness of C_a is standard (see e.g. Lemma 29 of [3]). To see that S_a is meager for non-ITRM-computable a, observe that the Cohen-generic reals over $L_{\omega_\omega^{\mathrm{CK},a}+1}[a]$ form a comeager set of reals to none of which a is reducible. $\square$

Definition 13. Let $x, y \subseteq \omega$. If x is ITRM-c-random relative to y and y is ITRM-c-random relative to x, we say that x and y are mutually ITRM-c-random.

Intuitively, we should expect that mutually random reals have no non-trivial information in common. This is expressed by the following theorem:

Theorem 14. If z is ITRM-computable from two mutually ITRM-c-random reals x and y, then z is ITRM-computable.

Proof. Assume otherwise, and suppose that z, x and y constitute a counterexample. By assumption, z is computable from x. Also, by Theorem 5, let P be a program such that $P^a(i)\downarrow$ for every $a \subseteq \omega$, $i \in \omega$ and such that P computes z in the oracle y. In the oracle z, the set $A_z := \{a | \forall i \in \omega P^a(i) \downarrow= z(i)\}$ is decidable by simply computing $P^a(i)$ for all $i \in \omega$ and comparing the result to the ith bit of z. Clearly, we have $A_z \subseteq \{a \mid z \leq_{\mathrm{ITRM}} a\}$. Hence, by our Lemma 12 above, A_z is meager as z is not ITRM-computable by assumption. Since A_z is decidable in the oracle z and z is computable from x, A_z is also decidable in the oracle x. Now, x and y are mutually ITRM-c-random, so that $y \notin A_z$. But P computes z in the oracle y, so $y \in A_z$ by definition, a contradiction. $\square$

While, naturally, there are non-computable reals that are reducible to a c-random real x (such as x itself), intuitively, it should not be possible to compute a non-computable real that is 'unique' is some effective sense from a random real. We approximate this intuition by taking 'unique' to mean 'ITRM-recognizable' (see [10], [1] or [2] for more information on ITRM-recognizability). It turns out that, in accordance with this intuition, recognizables that are ITRM-computable from ITRM-c-random reals are already ITRM-computable.

Definition 15. $x \subseteq \omega$ is ITRM-recognizable if and only if $\{x\}$ is ITRM-decidable. RECOG denotes the set of recognizable reals.

Theorem 16. Let $x \in \text{RECOG}$ and let y be ITRM-c-random such that $x \leq_{\text{ITRM}} y$. Then x is ITRM-computable.

Proof. Let $x \in \text{RECOG} \setminus \text{COMP}$ be computable from y, say by program P, and let Q be a program that recognizes x. The set $S := \{z \mid P^z \downarrow= x\}$ is meager as in the proof of Theorem 14. But S is decidable: Given a real z, use a halting-problem solver for P (which exists uniformly in the oracle by Theorem 5) to test whether $P^z(i) \downarrow$ for all $i \in \omega$; if not, then $z \notin S$. Otherwise, use Q to check whether the real computed by P^z is equal to x. If not, then $z \notin S$, otherwise $z \in S$. As P^y computes x, it follows that $y \in S$, so that y is an element of an ITRM-decidable meager set. Hence y is not ITRM-c-random, a contradiction. $\square$

Remark: Let $(P_i | i \in \omega)$ be a natural enumeration of the ITRM-programs. Together with the fact that the halting number $h = \{i \in \omega \mid P_i \downarrow\}$ for ITRMs is recognizable (see [2]), this implies in particular that the halting problem for ITRMs is not ITRM-reducible to an ITRM-c-random real. In particular, the Kucera-Gacs theorem, which says that every real is reducible to a random real (see e.g. Theorem 8.3.2 of [6]), does not hold in our setting.

3 An analogue to van Lambalgen's theorem

A crucial result of classical algorithmic randomness is van Lambalgen's theorem, which states that for reals a and b, $a \oplus b$ is ML-random if and only if a is ML-random and b is ML-random relative to a. In this section, we demonstrate an analogous result for ITRM-c-randomness. This will be a crucial ingredient in our considerations on ITRM-degrees below.

Lemma 17. Let Q be a deciding ITRM-program using n registers and $a \subseteq \omega$. Then $\{y | Q^{y \oplus a} \downarrow= 1\}$ is meager if and only if $Q^{x \oplus a} \downarrow= 0$ for all $x \in L_{\omega_{n+1}^{\text{CK},a}+3}[a]$ that are Cohen-generic over $L_{\omega_{n+1}^{\text{CK},a}+1}[a]$.

Proof. By absoluteness of computations and the bound on ITRM-halting times (see Theorem 5), $Q^{x \oplus a} \downarrow= 0$ implies that $Q^{x \oplus a} \downarrow= 0$ also holds in $L_{\omega_{n+1}^{\text{CK},a}}[a]$. As this is expressible by a Σ_1-formula, it must be forced by some condition p by the forcing theorem over KP (see e.g. Theorem 10.10 of [16]).

Hence every Cohen-generic y extending p will satisfy $Q^{y \oplus a} \downarrow= 0$. The set C of reals Cohen-generic over $L_{\omega_{n+1}^{\text{CK},a}+1}[a]$ is comeager. Hence, if $Q^{x \oplus a} \downarrow= 0$ for some $x \in C$, then $Q^{x \oplus a} \downarrow= 0$ for a non-meager (in fact comeager in some interval) set C'. Now, for each condition p, $L_{\omega_{n+1}^{\text{CK},a}+3}[a]$ will contain a generic filter over $L_{\omega_{n+1}^{\text{CK},a}+1}[a]$ extending p (as $L_{\omega_{n+1}^{\text{CK},a}+1}[a]$ is countable in $L_{\omega_{n+1}^{\text{CK},a}+3}[a]$).

Hence, if $Q^{x\oplus a}\downarrow=0$ for all $x\in C\cap L_{\omega_{n+1}^{\text{CK},a}+3}[a]$, then this holds for all elements of C and the complement $\{y|Q^{y\oplus a}\downarrow=1\}$ is therefore meager.

If, on the other hand, $Q^{x\oplus a}\downarrow=1$ for some such x, then this already holds for all x in some non-meager (in fact comeager in some interval) set C' by the same reasoning. $\square$

Corollary 18. For a deciding ITRM-program Q using n registers, there exists an ITRM-program P such that, for all $x,y\in\mathfrak{P}(\omega)$, $P^{x}\downarrow=1$ if and only if $\{y|Q^{x\oplus y}\downarrow=1\}$ is non-meager and $P^{x}\downarrow=0$, otherwise.

Proof. From x, compute, using sufficiently many extra registers, the $<_L$-minimal real code $c:=\text{cc}(L_{\omega_{n+1}^{\text{CK},x}+3}[x])$ for $L_{\omega_{n+1}^{\text{CK},x}+3}[x]$ in the oracle x. This can be done uniformly in x. Then, using c, one can use the recursive algorithm developed in in section 6 of [9] to evaluate statements in $L_{\omega_{n+1}^{\text{CK},x}+3}[x]$. Hence, we can search through c, identify all elements which code reals $y\in L_{\omega_{n+1}^{\text{CK},x}+3}[x]$ that are Cohen-generic over $L_{\omega_{n+1}^{\text{CK},x}+1}$ and run $Q^{x\oplus y}$ for each of them to see whether $Q^{x\oplus y}\downarrow=1$; if yes, then we return 1, otherwise, we return 0. $\square$

Corollary 19. Let x,y be real numbers. Then x is ITRM-c-random in the oracle y if and only if x is Cohen-generic over $L_{\omega_{\omega}^{\text{CK},y}}[y]$.

Proof. Let S denote the set of Cohen-generic reals over $L_{\omega_{\omega}^{\text{CK},y}}[y]$. Then $x\in S$ if and only if $x\cap D\neq\emptyset$ for every dense subset $D\in L_{\omega_{\omega}^{\text{CK},y}}[y]$ of Cohen-forcing. Clearly, for every such D, $G_D:=\{y\mid y\cap D\neq\emptyset\}$ is comeager and ITRM-decidable in the oracle y (and so its complement is ITRM-decidable in y and meager), so every real number that is ITRM-c-random real relative to y must be contained in every G_D and hence also in S.

On the other hand, if $x\in S$ and $P^{x\oplus y}\downarrow=1$ for some ITRM-program P that is deciding in the oracle y, then there is some finite $p\subseteq x$ such that $P^{z\oplus y}\downarrow=1$ for every $p\subset z\in S$, so the set decided by P in y is not meager. Hence x is not an element of any meager set ITRM-decidable in the oracle y, so x is ITRM-c-random relative to y. $\square$

We now give an ITRM-analogue of van Lambalgen's theorem and prove it following the strategy used in the classical setting, see e.g. [6], Theorem 6.9.1 and 6.9.2.

Theorem 20. Assume that a and b are reals such that $a\oplus b$ is not ITRM-c-random. Then a is not ITRM-c-random or b is not ITRM-c-random relative to a.

Proof. As $a \oplus b$ is not ITRM-c-random, let X be an ITRM-decidable meager set of reals such that $a \oplus b \in X$. Suppose that P is a program deciding X.

Let $Y := \{x | \{y \mid x \oplus y \in X\}$ non-meager$\}$. By Corollary 18, Y is ITRM-decidable.

We claim that Y is meager. First, Y is provably Δ^1_2 and hence has the Baire property (see e.g. Exercise 14.4 of [12]). Hence, by the Kuratowski-Ulam-theorem (see e.g. [13], Exercise 5A.9), Y is meager. Consequently, if $a \in Y$, then a is not ITRM-c-random.

Now suppose that $a \notin Y$. This means that $\{y \mid a \oplus y \in X\}$ is meager. But $S := \{y \mid a \oplus y \in X\}$ is easily seen to be ITRM-decidable in the oracle a and $b \in S$. Hence b is not ITRM-c-random relative to a. □

Theorem 21. Assume that a and b are reals such that $a \oplus b$ is ITRM-c-random. Then a is ITRM-c-random and b is ITRM-c-random relative to a.

Proof. Assume first that a is not ITRM-c-random, and let X be an ITRM-decidable meager set with $a \in X$. Then $X \oplus [0, 1]$ is also meager and ITRM-decidable. As $a \in X$, we have $a \oplus b \in X \oplus [0, 1]$, so $a \oplus b$ is not ITRM-c-random, a contradiction.

Now suppose that b is not ITRM-c-random relative to a, and let X be a meager set of reals such that $b \in X$ and X is ITRM-decidable in the oracle a. Let Q be an ITRM-program such that Q^a decides X. Our goal is to define a deciding program $\tilde{Q}$ such that $\tilde{Q}^a$ still decides X, but also $\{x | \tilde{Q}^x \downarrow= 1\}$ is meager. This suffices, as then $\tilde{Q}^{a\oplus b} \downarrow= 1$ and $\{x | \tilde{Q}^x \downarrow= 1\}$ is ITRM-decidable, so that $a \oplus b$ cannot be ITRM-c-random. $\tilde{Q}$ operates as follows: Given $x = y \oplus z$, check whether $\{w \mid Q^{y\oplus w} \downarrow= 1\}$ is meager, using Corollary 18. If that is the case, carry out the computation of Q^x and return the result. Otherwise, return 0. This guarantees (since X is meager) that $\{z \mid \tilde{Q}^{y\oplus z} \downarrow= 1\}$ is meager and furthermore that $\tilde{Q}^{a\oplus z} \downarrow= 1$ if and only if $Q^{a\oplus z} \downarrow= 1$ if and only if $z \in X$ for all reals z, so that $\{z | \tilde{Q}^{a\oplus z} \downarrow= 1\}$ is just X, as desired. □

Combining Theorem 20 and 21 gives us the desired conclusion:

Theorem 22. Given reals x and y, $x \oplus y$ is ITRM-c-random if and only if x is ITRM-c-random and y is ITRM-c-random relative to x. In particular, if x and y are ITRM-c-random, then x is ITRM-c-random relative to y if and only if y is ITRM-c-random relative to x.

Remark: Combined with Corollary 19, this shows 'computationally' that, over $L_{\omega_\omega^{\mathrm{CK}}}$, $x \oplus y$ is Cohen-generic if and only if x is Cohen-generic over $L_{\omega_\omega^{\mathrm{CK}}}$ and y is Cohen-generic over $L_{\omega_\omega^{\mathrm{CK}}}[x]$.

We note that a classical corollary to van Lambalgen's theorem continues to hold in our setting:

Corollary 23. Let x, y be ITRM-c-random. Then x is ITRM-c-random relative to y if and only if y is ITRM-c-random relative to x.

Proof. Assume that y is ITRM-c-random relative to x. By assumption, x is ITRM-c-random. By Theorem 22, $x \oplus y$ is ITRM-c-random. Trivially, $y \oplus x$ is also ITRM-c-random. Again by Theorem 22, x is ITRM-c-random relative to y. By symmetry, the corollary holds. $\square$

Based on Theorem 16, one might conjecture that a real x which is ITRM-computable from an ITRM-c-random real y must be computable or itself ITRM-c-random. This, however does not hold:

Theorem 24. For every ITRM-c-random real y, there is a real x such that x is neither ITRM-c-random nor ITRM-computable and x is ITRM-computable from y. However, if x is ITRM-computable from an ITRM-c-random real y and x is an element of an ITRM-decidable set S such that $\text{card}(S) = \aleph_0$, then x is computable.

Proof. Let $x := 0 \oplus y$, then x is clearly computable from y. However, x is not computable as the non-computable y is computable from x. Furthermore, x is contained in the ITRM-decidable meager set $\{0\} \oplus [0, 1]$, so x cannot be ITRM-c-random.

For the second statement, assume for a contradiction that S is an ITRM-decidable countable set containing a non-computable real x such that $x \leq_{\text{ITRM}} y$ for some ITRM-c-random real y. Let P be a program deciding S and let Q be a program such that $Q^y \downarrow= x$. By Theorem 5, we may assume without loss of generality that Q^z computes some real for every $z \subseteq \omega$. By $\bar{S}$, we denote the set of $z \in S$ such that $M_a := \{a | Q^a \downarrow= z\}$ is meager, $Q(z)$ denotes the real number computed by Q^z. Then, as x is not ITRM-computable, we have $x \in \bar{S}$ by Theorem 9. Furthermore $\bar{S}$ is decidable: For let R be a program such that $R^{a\oplus b} \downarrow= 1$ if and only if $Q^a \downarrow= b$ and $R^{a\oplus b} \downarrow= 0$ otherwise; so R is deciding. Now $a \in \bar{S}$ if and only if $a \in S$ and $\{b | R^{a\oplus b}\}$ is meager, hence, by Theorem 18, $\bar{S}$ is decidable by some program $\bar{P}$. Let $M := \{z | \bar{P}^{Q(z)} \downarrow= 1\}$ be the set of all reals z such that $Q(z) \in \bar{S}$. Then M is obviously ITRM-decidable. As $Q(y) = x \in \bar{S}$, we have $y \in M$. As y is ITRM-c-random, it follows that M is not meager. But $M = \bigcup_{a\in S} M_a$; as S is countable, there must thus be some $a \in S$ such that M_a is not meager, a contradiction. $\square$

Note that the condition of being ITRM-computable from an ITRM-c-random real is necessary in the assumption of Theorem 24: It is not true

that every element of a countable ITRM-decidable set is ITRM-computable; in fact, there are countable ITRM-decidable sets that do not contain any ITRM-computable element:

Proposition 25. The set $X := \{x \subseteq \omega : \exists \alpha \in \mathrm{On}(\omega_{\omega}^{\mathrm{CK}} < \alpha < \omega_{2\omega}^{\mathrm{CK}} \wedge$ 'x is the $<_L$-minimal real in $L_{\alpha+1} \setminus L_{\alpha}$ coding L_{α}')$\}$ is countable, ITRM-decidable and contains no ITRM-computable element.

Proof. As $\omega_{2\omega}^{\mathrm{CK}}$ is countable, the set of indices between $\omega_{\omega}^{\mathrm{CK}}$ and $\omega_{2\omega}^{\mathrm{CK}}$ is countable and hence so is the set of the corresponding codes for L-levels. Also, by Theorem 4, all ITRM-computable reals are contained in $L_{\omega_{\omega}^{\mathrm{CK}}}$, so X has no ITRM-computable element.

To see that X is ITRM-decidable, we apply the techniques from section 6 of [10] to check, for a given oracle x, whether x codes an L-level L_{α} and whether this L_{α} contains exactly one limit of admissible ordinals. It is also shown in [10] how to uniformly compute a code c for $L_{\alpha+1}$ from x when x codes L_{α}. Hence, it only remains to check whether $L_{\alpha+1} \setminus L_{\alpha}$ contains x and whether x is $<_L$-minimal in $L_{\alpha+1}$ with these properties, and the answers to both questions can be easily read off from c. $\square$

Remark: We do not know whether a countable ITRM-decidable set X can contain a non-ITRM-recognizable element. Our conjecture is that this is not possible and that in fact any ITRM-decidable set with a non-ITRM-recognizable element must have a perfect subset (and hence have cardinality $2^{\aleph_0}$). The next result, however, shows that every non-empty ITRM-decidable set (countable or not) contains **some** recognizable element. This can be seen as a kind of basis theorem for ITRMs:

Theorem 26. Every non-empty ITRM-decidable set of real numbers has a recognizable element.

Proof. Let $X \neq \emptyset$ be ITRM-decidable, and let P be a program that decides X. Let $\Gamma := \{\omega_{\omega}^{\mathrm{CK},x} : x \in X\}$. Since X is non-empty, so is Γ; let μ be the minimal element of Γ and pick $x_{\gamma} \in X$ such that $\omega_{\omega}^{\mathrm{CK},x_{\gamma}} = \mu$. By Theorem 5, we thus have $P^{x_{\gamma}} \downarrow = 1$ in less than μ many steps. By absoluteness of computations, it follows that $L_{\mu}[x_{\gamma}] \models P^{x_{\gamma}} \downarrow = 1$ and hence $L_{\mu}[x_{\gamma}] \models \exists x P^{x} \downarrow = 1$. By the Jensen-Karp theorem (see [11]), the Σ_1-formula $\exists x P^{x} \downarrow = 1$ is absolute between $L_{\mu}[x_{\gamma}]$ and L_{μ}; hence $L_{\mu} \models \exists x P^{x} \downarrow = 1$. Let $\bar{x} \in L_{\mu}$ be the $<_L$-minimal element of L_{μ} such that $L_{\mu} \models P^{\bar{x}} \downarrow = 1$; thus $\bar{x} \in X$ since $P^{\bar{x}} \downarrow = 1$ by absoluteness of computations.

We claim that $\bar{x}$ is recognizable. To see this, first note that $\bar{x} \in L_{\omega_{\omega}^{\mathrm{CK},\bar{x}}}$, so that there is $k \in \omega$ with $\bar{x} \in L_{\omega_{k}^{\mathrm{CK},\bar{x}}}$. Suppose that P uses n registers and

let $m = \max\{k, n+1\}$. Then the Σ_1-hull of $\{\bar{x}\}$ in $L_{\omega_m^{\mathrm{CK},\bar{x}}}$ will be isomorphic (via the collapsing map) to $L_{\omega_m^{\mathrm{CK},\bar{x}}}$, and so $L_{\omega_m^{\mathrm{CK},\bar{x}}+1}$ will contain a real number r coding $L_{\omega_m^{\mathrm{CK},\bar{x}}}$ by standard fine-structural arguments. Hence r is ITRM-computable from $\bar{x}$; let Q be an ITRM-program computing r from $\bar{x}$. To recognize $\bar{x}$, we now proceed as follows: Let y be given in the oracle. First test whether $P^y \downarrow= 1$. If not, then $y \neq \bar{x}$. Otherwise, test, using Q and a halting problem solver for Q, whether Q^y computes a real r coding an admissible L_α containing y. If not, then $y \neq \bar{x}$. If yes, then use r to search for an element z of L_α that is $<_L$-smaller than y and satisfies $P^z \downarrow= 1$. If the search is successful, then $y \neq \bar{x}$. On the other hand, if none is found, then y is the $<_L$-smallest element a of X with $a \in L_{\omega_\omega^{\mathrm{CK},a}}$, so $y = \bar{x}$. $\square$

4 Some consequences for the structure of ITRM-degrees

In the new setting, we can also draw some standard consequences of van Lambalgen's theorem.

Definition 27. If $x \leq_{\mathrm{ITRM}} y$ but not $y \leq_{\mathrm{ITRM}} x$, we write $x <_{\mathrm{ITRM}} y$. If $x \leq_{\mathrm{ITRM}}$ and $y \leq_{\mathrm{ITRM}} x$, then we write $x \equiv_{\mathrm{ITRM}} y$. If neither $x \leq_{\mathrm{ITRM}} y$ nor $y \leq_{\mathrm{ITRM}} x$, we call x and y incomparable and write $x \perp_{\mathrm{ITRM}} y$.

Clearly, $\equiv_{\mathrm{ITRM}}$ is an equivalence relation. We may hence form, for each real x, the $\equiv_{\mathrm{ITRM}}$-equivalence class $[x]_{\mathrm{ITRM}}$ of x, called the ITRM-degree of x. It is easy to see that $\leq_{\mathrm{ITRM}}$ respects $\equiv_{\mathrm{ITRM}}$, so that expressions like $[x]_{\mathrm{ITRM}} \leq_{\mathrm{ITRM}} [y]_{\mathrm{ITRM}}$ etc. are well-defined and have the obvious meaning.

Corollary 28. If a is ITRM-c-random, $a = a_0 \oplus a_1$, then $a_0 \perp_{\mathrm{ITRM}} a_1$.

Proof. By Theorem 22, a_0 and a_1 are mutually ITRM-c-random. If a_0 was ITRM-computable from a_1, then $\{a_0\}$ would be decidable in the oracle a_1, meager and contain a_0, so a_0 would not be ITRM-c-random relative to a_1, a contradiction. By symmetry, the claim follows. $\square$

Lemma 29. Let h be a real coding the halting problem for ITRMs as in the remark following Theorem 16. Then there is an ITRM-c-random real $x \leq_{\mathrm{ITRM}} h$.

Proof. Given h, we can compute a code for $L_{\omega_\omega^{\mathrm{CK}}+2}$ as follows: By Theorem 4.6 of [2], h is ITRM-recognizable. Clearly, h is not ITRM-computable. Hence, by Theorem 3.2 of [2], we have $\omega_\omega^{\mathrm{CK},h} > \omega_\omega^{\mathrm{CK}}$. By Theorem 4, a real r is ITRM-computable from h if and only if it is an element of $L_{\omega_\omega^{\mathrm{CK},h}}[h]$. As $L_{\omega_\omega^{\mathrm{CK}}+3}$ contains a code for $L_{\omega_\omega^{\mathrm{CK}}+2}$, such a code is contained in $L_{\omega_\omega^{\mathrm{CK},h}} \subseteq L_{\omega_\omega^{\mathrm{CK},h}}[h]$ and is hence ITRM-computable from h.

Now $L_{\omega_\omega^{\mathrm{CK}+2}}$ will contains a real x which is Cohen-generic over $L_{\omega_\omega^{\mathrm{CK}}+1}$. Such an x can easily be computed from a code for $L_{\omega_\omega^{\mathrm{CK}+2}}$ and consequently, x itself is ITRM-computable from h. By Corollary 19, x is ITRM-c-random. $\square$

We have an analogue to the Kleene-Post-theorem on Turing degrees between 0 and $0'$ (see e.g. Theorem VI.1.2 of [20]) for ITRMs.

Corollary 30. With h as in Lemma 29, there are x_0, x_1 such that we have $[0]_{\mathrm{ITRM}} <_{\mathrm{ITRM}} [x_0]_{\mathrm{ITRM}}, [x_1]_{\mathrm{ITRM}} \leq h$ and $x_0 \perp_{\mathrm{ITRM}} x_1$. In particular, there is a real x_0 such that $[0]_{\mathrm{ITRM}} <_{\mathrm{ITRM}} [x_0]_{\mathrm{ITRM}} <_{\mathrm{ITRM}} h$.

Proof. Pick x as in Lemma 29, let $x = x_0 \oplus x_1$, and use Corollary 28. $\square$

Using van Lambalgen, we can show much more:

Theorem 31. Let $\mathfrak{T}$ denote the tree of finite 0-1-sequences, ordered by reverse inclusion. Then $\mathfrak{T}$ embeds in the ITRM-degrees between 0 and $0'$, i.e. there is an injection $f : \mathfrak{T} \to \mathfrak{P}(\omega)$ such that, for all $x, y \in \mathfrak{T}$, we have $0 <_{\mathrm{ITRM}} f(x), f(y) <_{\mathrm{ITRM}} 0'_{\mathrm{ITRM}}$ and $f(x) \leq_{\mathrm{ITRM}} f(y)$ if and only if $x \supseteq y$. In particular, there is an infinite descending sequence of ITRM-degrees between 0 and $0'_{\mathrm{ITRM}}$.

Proof. We shall, for each $s \in^{<\omega} 2$, define a real r_s in such a way that the set $\{[r_s]_{\mathrm{ITRM}} | s \in^{<\omega} 2\}$ has the desired property. Let $r = r_{()}$ be ITRM-c-random such that $0 <_{\mathrm{ITRM}} r <_{\mathrm{ITRM}} 0'_{\mathrm{ITRM}}$. Assuming that r_s is already defined, let $r_s = x_0 \oplus x_1$ and set $r_{s0} = x_0$, $r_{s1} = x_1$. Clearly, if s is a proper initial segment of t, then $r_s <_{\mathrm{ITRM}} r_t$. We show that furthermore, when s and t are incompatible (i.e. none is an initial segment of the other), then $r_s \perp_{\mathrm{ITRM}} r_t$: Let $u = r_s \cap r_t$ be the longest common initial segment of r_s and r_t. Without loss of generality, assume that s starts with $u0$ and that t starts with $u1$. Then $r_u = r_{u0} \oplus r_{u1}$, so, by Theorem 22, $u0$ and $u1$ are mutually random. Now suppose that r_s and r_t are not mutually random, e.g. that r_s was not random relative to r_t. Then, as r_t is recursive in r_{u1}, r_s is not random relative to r_{u1}. Let U be a meager set containing r_s which is ITRM-decidable in the oracle r_{u1}.

Let s' be a binary string such that $u0s' = s$ and let l denote its length. We recursively define a sequence $(U_i : 0 \leq i \leq l)$ of subsets of $[0,1]$ as follows: Let $U_0 = 0$ and for $\leq i < n$ let $U_{i+1} := U_i \oplus [0,1]$ when the $l-i$th digit of l is 0 and otherwise $U_{i+1} := [0,1] \oplus U_i$. By an easy induction, we have that $r_{u0(s'|i)} \in U_{l-i}$ for $0 \leq i \leq l$, where $s'|i$ denotes the restriction of s' to its first i many bits. In particular, we have $r_{u0} \in U_l$. Moreover, when X is meager, then so are $X \oplus [0,1]$ and $[0,1] \oplus X$, so as $U_0 = U$ is meager, it follows by

another induction that U_i is meager for $0 \leq i \leq l$ and in particular that U_l is meager. It is also easy to see inductively that, as $U_0 = U$ is ITRM-decidable in the oracle r_{u1}, so is U_i for all $0 \leq i \leq l$ and hence in particular for $i = l$. So U_l is meager, contains r_{u0} and is ITRM-decidable in the oracle r_{u1}, contradicting the fact that r_{u0} is ITRM-c-random relative to r_{u1}. $\square$

Corollary 32. The ITRM-degrees of ITRM-c-random reals contain no minimal or maximal element under ITRM-reduction.

Proof. Let y is ITRM-c-random, $y = y_1 \oplus y_2$.

Then first, by Theorem 22, y_1 is ITRM-c-random and $y_1 <_{\text{ITRM}} y$. Hence there is no $<_{\text{ITRM}}$-minimal ITRM-c-random real y.

Furthermore, we have $\omega_\omega^{\text{CK},y} = \omega_\omega^{\text{CK}}$ by Theorem 10, so $L_{\omega_\omega^{\text{CK},y}}[y] = L_{\omega_\omega^{\text{CK}}}[y]$; Let x be Cohen-generic over $L_{\omega_\omega^{\text{CK}}}[y]$, then, by the relativized version of Lemma 19 (see the remark following the Lemma), x is ITRM-c-random relative to y. By Theorem 22, $z := x \oplus y$ is ITRM-c-random, and we clearly have $y <_{\text{ITRM}} z$, so y is not maximal.

$\square$

In the Turing degrees, it is a striking experience that intermediate degrees (i.e. degrees lying properly between two iterations of the Turing jump) don't seem to come up 'naturally', but need to be constructed on purpose (see e.g. the discussion in [18]). Concerning ITRMs, we can to a certain extent prove that something similar is going on: Namely, reals of intermediate degree are never recognizable.

Theorem 33. If $x \subseteq \omega$ is recognizable, then $x \in L_{\omega_\omega^{\text{CK},x}}$.

Proof. Let P be a program recognizing x. Then, by Theorem 5, P^x stops in less than $\omega_\omega^{\text{CK},x}$ many steps. Consequently, the computation is contained in $V_{\omega_\omega^{\text{CK},x}}$. Hence $V_{\omega_\omega^{\text{CK},x}} \models \exists y P^y \downarrow= 1$. Clearly, $\omega_\omega^{\text{CK},x}$ is a limit of admissible ordinals. By a theorem of Jensen and Karp ([11]), Σ_1-formulas are absolute between L_α and V_α whenever α is a limit of admissibles. Hence $L_{\omega_\omega^{\text{CK},x}} \models \exists y P^y \downarrow= 1$. By absoluteness of computations then, we have $x \in L_{\omega_\omega^{\text{CK},x}}$, as desired. $\square$

Lemma 34. If x is recognizable, then so is $x \oplus x'_{\text{ITRM}}$.

Proof. This is a relativized version of Theorem 26 of [2]. We sketch the proof for completeness: By assumption, let P be an ITRM-program that recognizes x.

It then follows that $\omega_\omega^{\text{CK},x'} > \omega_\omega^{\text{CK},x}$: The idea is to first use x'_{ITRM} to decide for any program P_i whether P_i^x computes a real coding a well-ordering;

moreover, these codes can be read out explictly with the help of x'. To see this, let P_i be given. It is easy to compute from i the index $f(i)$ for an ITRM-program $P_{f(i)}$ such that $P^x_{f(i)}$ halts if and only if P^x_i halts with output 0 or 1, i.e. computes a real number. Given that P^x_i computes a real number, as ITRMs can check relations for well-foundedness (see [9]), it is also easy to effectively obtain from i an ITRM-program index $g(i)$ such that $P^x_{g(i)}$ halts if and only if the real computed by P^x_i codes a well-founded relation. Furthermore, given that P^x_i computes a real number, it is also easy to effectively obtain from i an ITRM-program index $h(i)$ such that for every $j \in \omega$, $P^x_{h(i)}(j)$ halts if and only if the jth bit of the real number computed by P^x_i is 1. Hence, using x', we can compute the jth bit of the ith code of a well-ordering ITRM-computable in the oracle x.

Finally, we can now assemble all of these codes into a code for a well-ordering of order type their sum, which will be the supremum of the ordinals with codes ITRM-computable in the oracle x, i.e. $\omega_\omega^{\mathrm{CK},x}$. Thus, from x', we can compute a code for a well-ordering of order-type $\omega_\omega^{\mathrm{CK},x}$.

As $\omega_\omega^{\mathrm{CK},x'}$ is the supremum of the ordinals with codes ITRM-computable in the oracle x', it follows that $\omega_\omega^{\mathrm{CK},x'} > \omega_\omega^{\mathrm{CK},x}$; hence there is some minimal $k \in \omega$ such that $\omega_k^{\mathrm{CK},x'} > \omega_\omega^{\mathrm{CK},x}$. There is a real c coding $L_{\omega_\omega^{\mathrm{CK},x}}[x]$ contained in $L_{\omega_\omega^{\mathrm{CK},x}+1}[x]$ and hence in $L_{\omega_k^{\mathrm{CK},x'}}[x']$ (as x is computable from x'). By Theorem 4 above, there is an ITRM-program Q computing c in the oracle x'. But as $\omega_\omega^{\mathrm{CK},x}$ is the supremum of the halting times for ITRM-programs in the oracle x, a program in the oracle x will halt inside $L_{\omega_\omega^{\mathrm{CK},x}}[x]$ if and only if it halts in the real world. Checking whether some ITRM-program P^x_i halts hence reduces to checking a first-order statement in the structure coded by c, which can be done as described in the last section of [10]. x' can now be identified by checking whether the statement '$P^x_i\downarrow$' holds in the structure coded by c for each $i \in \omega$ and comparing the results to the oracle.

This leads to the following procedure for recognizing $x \oplus x'$: Suppose that $y = y_1 \oplus y_2$ is given in the oracle. First, we run P on y_1 to check whether $y_1 = x$. If not, then $y \neq x \oplus x'$. Otherwise, check, using a halting problem solver for Q, whether Q^{y_2} computes a real r coding $L_{\omega_\omega^{\mathrm{CK},y_1}}[y_1]$. If not, then $y \neq x \oplus x'$. Otherwise, using r, check for each $i \in \omega$ whether $L_{\omega_\omega^{\mathrm{CK},x}}[x] \models P^x_i \downarrow \leftrightarrow i \in y_2$. If not, then $y \neq x \oplus x'$, otherwise we have $y = x \oplus x'$. $\square$

Corollary 35. If $x \subseteq \omega$ is recognizable, then so is x'_{ITRM}.

Proof. Note that x is Turing-computable from x'_{ITRM}: To determine the ith bit of x when x'_{ITRM} is given, simply use x'_{ITRM} to check whether the ITRM-program Q_i that stops when the ith bit of its oracle is 1 and otherwise enters

an infinite loop halts in the oracle x. This works uniformly in x. Let Q be an ITRM-program that computes x from x'_{ITRM}, for every $x \subseteq \omega$.

To recognize x'_{ITRM}, we proceed as follows: First, let P be a program that recognizes x. Now, given y in the oracle, first check, using a halting problem solver for Q, whether Q^y computes a real number. If not, then $y \neq x'$. Otherwise, let z be that number and run P^z. If $P^z \downarrow = 0$, then $z \neq x$, so $y \neq x'_{\mathrm{ITRM}}$ (since Q would have computed x from x'_{ITRM}). Otherwise, we have $z = x$ and can use Lemma 34 to check whether $z \oplus y = x \oplus x'_{\mathrm{ITRM}}$, which will be the case if and only if $y = x'_{\mathrm{ITRM}}$. □

Theorem 36. For $i \in \omega$, we have $\omega_\omega^{\mathrm{CK},0^{(i)}_{\mathrm{ITRM}}} = \omega^{\mathrm{CK}}_{\omega(i+1)}$; consequently, $i+1$ is the minimal $n \in \omega$ such that $0^{(i)}_{\mathrm{ITRM}} \in L_{\omega^{\mathrm{CK}}_{\omega n}}$.

Proof. For $i \in \omega$, let σ_i denote $\omega_\omega^{\mathrm{CK},0^{(i)}_{\mathrm{ITRM}}}$.

It is easy to see that $0^{(i)}_{\mathrm{ITRM}}$ is ITRM-computable from $0^{(i+1)}_{\mathrm{ITRM}}$ for all $i \in \omega$. Hence, by Lemma 34, $0^{(i)}_{\mathrm{ITRM}}$ is recognizable for each $i \in \omega$. By Theorem 33 then, $0^{(i)}_{\mathrm{ITRM}} \in L_{\sigma_i}$ for each $i \in \omega$; thus $L_{\sigma_i}[0^{(i)}_{\mathrm{ITRM}}] = L_{\sigma_i}$. Hence a real x is computable from $0^{(i)}_{\mathrm{ITRM}}$ if and only if $x \in L_{\sigma_i}$. As $0^{(i+1)}_{\mathrm{ITRM}}$ is not ITRM-computable from $0^{(i)}_{\mathrm{ITRM}}$, we must have $\omega_\omega^{\mathrm{CK},0^{(i+1)}_{\mathrm{ITRM}}} > \omega_\omega^{\mathrm{CK},0^{(i)}_{\mathrm{ITRM}}}$ for all $i \in \omega$; hence $(\sigma_i | i \in \omega)$ is a strictly increasing sequence of limits of admissible ordinals.

We now proceed inductively to show that $\sigma_i = \omega^{\mathrm{CK}}_{\omega(i+1)}$:

As $0'_{\mathrm{ITRM}}$ is not ITRM-computable, it is not an element of $L_{\omega^{\mathrm{CK}}_\omega}$. As it is definable over $L_{\omega^{\mathrm{CK}}_\omega}$, it is an element of $L_{\omega^{\mathrm{CK}}_\omega+1}$ and hence of $L_{\omega^{\mathrm{CK}}_{\omega2}}$. Hence $\omega^{\mathrm{CK}}_\omega < \sigma_1 \leq \omega^{\mathrm{CK}}_{\omega2}$; as σ_1 is a limit of admissibles, we have $\sigma_1 = \omega^{\mathrm{CK}}_{\omega2}$.

Similarly, assuming that $\sigma_i = \omega^{\mathrm{CK}}_{\omega(i+1)}$, $0^{(i+1)}_{\mathrm{ITRM}}$ is definable over $L_{\omega^{\mathrm{CK}}_{\omega(i+1)}}$ and hence an element of $L_{\omega^{\mathrm{CK}}_{\omega(i+2)}}$. Hence $\omega^{\mathrm{CK}}_{\omega(i+1)} < \sigma_{i+1} \leq \omega^{\mathrm{CK}}_{\omega(i+2)}$, so $\sigma_{i+1} = \omega^{\mathrm{CK}}_{\omega(i+2)}$, as desired. □

We can now extend Theorem 31 to obtain intermediate degrees also for iterations of the jump operator:

Corollary 37. There is a real x such that $0'_{\mathrm{ITRM}} <_{\mathrm{ITRM}} x <_{\mathrm{ITRM}} 0''_{\mathrm{ITRM}}$.

Proof. Let $x \in L_{\omega^{\mathrm{CK}}_{\omega3}} - L_{\omega^{\mathrm{CK}}_{\omega2}}$ be Cohen-generic over $L_{\omega^{\mathrm{CK}}_{\omega2}}$ (and hence ITRM-c-random). (Such a real exists because, at $\omega^{\mathrm{CK}}_{\omega2}+1$, the ultimate projectum drops to ω, so that already $L_{\omega^{\mathrm{CK}}_{\omega2}+2}$ contains a real Cohen-generic over $L_{\omega^{\mathrm{CK}}_{\omega2}}$; this real is certainly Cohen-generic over $L_{\omega^{\mathrm{CK}}_\omega}$ and hence ITRM-c-random by Theorem 19, and cannot be an element of $L_{\omega^{\mathrm{CK}}_{\omega2}}$.)

So $x \in L_{\omega^{CK}_{\omega 3}} = L_{\omega_\omega^{CK,0''_{ITRM}}}[0''_{ITRM}]$ by Lemma 34 and Corollary 35, hence $x \leq_{ITRM} 0''_{ITRM}$. But we cannot have $x \in L_{\omega_\omega^{CK,0'_{ITRM}}}[0'_{ITRM}] = L_{\omega_\omega^{CK,0'_{ITRM}}}$ since $L_{\omega_\omega^{CK,0'_{ITRM}}} = L_{\omega^{CK}_{\omega 2}}$ by Theorem 36. Consequently, we have $x \leq 0''_{ITRM}$ and $x \not\leq 0'_{ITRM}$.

As x is Cohen-generic over $L_{\omega^{CK}_{\omega 2}}$ and $L_{\omega^{CK}_{\omega 2}} = L_{\omega_\omega^{CK,0'_{ITRM}}}[0'_{ITRM}]$ by Theorem 36, we have $\omega_\omega^{CK,x\oplus 0'_{ITRM}} = \omega_\omega^{CK,0'_{ITRM}} = \omega^{CK}_{\omega 2} < \omega^{CK}_{\omega 3} = \omega_\omega^{CK,0''_{ITRM}}$. In particular, there is a real coding a well-ordering of order type $\omega^{CK}_{\omega 3}$ ITRM-computable from $0''_{ITRM}$, but not from $x \oplus 0'_{ITRM}$. Hence $0''_{ITRM}$ cannot be ITRM-computable from $x \oplus 0'_{ITRM}$, so that $x \oplus 0'_{ITRM} \leq_{ITRM} 0''_{ITRM}$ implies $x \oplus 0'_{ITRM} <_{ITRM} 0''_{ITRM}$.

Hence $0'_{ITRM} <_{ITRM} x \oplus 0'_{ITRM} <_{ITRM} 0''_{ITRM}$, so $y := x \oplus 0'_{ITRM}$ is as desired. □

By a similar argument, one gets intermediate degrees between any two successive (finite) iterations of the ITRM-jump:

Corollary 38. For every $i \in \omega$, there is a real x_i such that $0^{(i)}_{ITRM} <_{ITRM} x < 0^{(i+1)}_{ITRM}$.

Given Corollary 38, one may ask whether any of these intermediate degrees is in some natural sense 'unique'. As above (see the passage preceeding Definition 15), a way to capture the meaning of 'unique' that suggests itself in the context of ITRMs is ITRM-recognizability. It turns out that, in this interpretation, no intermediate degree in the finite iterations of the ITRM-jump is 'unique':

Theorem 39. Assume that $x \leq_{ITRM} 0^{(i)}_{ITRM}$ for some $i \in \omega$ and that $x \in$ RECOG. Then there is $j \in \omega$ such that $x \in [0^{(j)}_{ITRM}]_{ITRM}$.

Proof. We start by showing: If $x >_{ITRM} 0^{(i)}_{ITRM}$ is recognizable, then $x \geq_{ITRM} 0^{(i+1)}_{ITRM}$. To see this, note that $x > 0^{(i)}_{ITRM}$ implies that $x \notin L_{\omega^{CK}_{\omega(i+1)}}$ by Theorem 36. As x is recognizable, we have $x \in L_{\omega_\omega^{CK,x}}$ by Theorem 33. Hence $\omega_\omega^{CK,x}$ is a limit of admissibles strictly bigger than $\omega^{CK}_{\omega(i+1)}$, so $\omega_\omega^{CK,x} \geq \omega^{CK}_{\omega(i+2)}$. As $0^{(i+1)}_{ITRM} \in L_{\omega^{CK}_{\omega(i+2)}}$, we have $0^{(i+1)}_{ITRM} \in L_{\omega_\omega^{CK,x}}[x]$, and hence $0^{(i+1)}_{ITRM} \leq x$. Now assume for a contradiction that x is as in the assumption of the theorem, but not ITRM-computably equivalent to some $0^{(i)}_{ITRM}$. In particular then, x is not ITRM-computable (otherwise $x \in [0^{(0)}_{ITRM}]_{ITRM}$). So $x >_{ITRM} 0^{(0)}_{ITRM}$. As $x \leq_{ITRM} 0^{(i)}_{ITRM}$ for some $i \in \omega$, there is some $j \in \omega$ such that $x \not>_{ITRM} 0^{(j)}$; without loss of generality, let j be minimal with this property. Then $j > 0$

and $x >_{\mathrm{ITRM}} 0^{(j-1)}_{\mathrm{ITRM}}$, so $x \geq_{\mathrm{ITRM}} 0^{(j)}_{\mathrm{ITRM}}$. As $x >_{\mathrm{ITRM}} 0^{(j)}_{\mathrm{ITRM}}$ is excluded, we have $x =_{\mathrm{ITRM}} 0^{(j)}_{\mathrm{ITRM}}$, so j is as desired. □

We can, however, also find reals x with degree strictly between 0 and $0'_{\mathrm{ITRM}}$ that are not random. This will, after some preparation, be shown in Theorem 44 below.

Definition 40. For $x, y \subseteq \omega$, we write $x \leq^n_{\mathrm{ITRM}} y$ to indicate that x is computable from y by an ITRM-program using at most n registers.

Lemma 41. There is a natural number C such that, for all reals x, y with $x \leq^n_{\mathrm{ITRM}} y$, we have $\omega_k^{CK,x} \leq \omega^{\mathrm{CK},y}_{Ck+n+1}$ for all $k \in \omega$.

Proof. It is not hard to see that there is a constant c such that, for every $0 < k \in \omega$ and every real z, a real coding $\omega_k^{\mathrm{CK},z}$ is (uniformly) ITRM-computable from z by a program P using at most kc many registers. Now, if $x \leq^n_{\mathrm{ITRM}} y$ via a program Q using at most n registers, we can, given y in the oracle, first use Q to compute x and then P to compute a code for $\omega_k^{\mathrm{CK},x}$. Hence, for some constant d (which is independent from k, x, y, n), we can compute a code for $\omega_k^{\mathrm{CK},x}$ in the oracle y using at most $kc + n + d$ many registers. With $C = c + d$, we hence get that such a code is computable from y with at most $kC + n$ many registers. However, every such real will be an element of $L_{\omega^{\mathrm{CK},y}_{kC+n+1}}[y]$ and hence cannot code an ordinal greater than $\omega^{\mathrm{CK},y}_{kC+n+1}$.Thus $\omega_k^{\mathrm{CK},x} \leq \omega^{\mathrm{CK},y}_{Ck+n+1}$. □

We recall the following special case of Theorem 4.1 from [19]:

Definition 42. An ordinal $\alpha < \omega_1^L$ is an index if and only if $L_{\alpha+1} - L_\alpha$ contains a real number.

Theorem 43. Let $(\alpha_\iota | \iota < \delta)$ be a countable sequence of admissible ordinals greater than ω such that, for each $\nu < \delta$, α_ν is admissible relative to $(\alpha_\iota | \iota < \nu)$. Then there is a real x such that α_ι is the ιth x-admissible ordinal greater than ω. With $\alpha = \sup\{\alpha_\iota | \iota < \delta\}$, such a real is arithmetically constructible from any real coding $(L_{\alpha+1}(\{\alpha_\iota | \iota < \delta\}), \in, \{\alpha_\iota | \iota < \delta\})$. If α is an index and $(\alpha_\iota | \iota < \delta)$ is definable over L_α, then such a real is contained in $L_{\alpha+2}$.

Theorem 44. There is a real $0 <_{\mathrm{ITRM}} x <_{\mathrm{ITRM}} 0'_{\mathrm{ITRM}}$ such that x is not ITRM-c-random; in fact, x can be chosen in such a way that x is not ITRM-reducible to any ITRM-c-random real y.

Proof. By Theorem 43, there is a real x such that $\omega_i^{\mathrm{CK},x} = \omega_{i^2}^{\mathrm{CK}}$ for all $i \in \omega$, definable over $L_{\omega_\omega^{\mathrm{CK}}+1}$ and hence an element of $L_{\omega_\omega^{\mathrm{CK}}+2}$. From $0'_{\mathrm{ITRM}}$, we can

compute a real coding $L_{\omega_\omega^{\mathrm{CK}}+2}$ and therefore also x, hence $x \leq_{\mathrm{ITRM}} 0'_{\mathrm{ITRM}}$. As $\omega_\omega^{\mathrm{CK},x} = \omega_\omega^{\mathrm{CK}} < \omega_\omega^{\mathrm{CK},0'}$ by Theorem 36, $0'_{\mathrm{ITRM}}$ is not computable from x: for otherwise, a code for $\omega_\omega^{\mathrm{CK}}$ would be computable from x. Now assume that x is reducible to an ITRM-c-random real y via a program P using n registers. By Corollary 10, we have $\omega_i^{\mathrm{CK},y} = \omega_i^{\mathrm{CK}}$ for all $i \in \omega$. By Lemma 41, there is $1 < C \in \omega$ such that $\omega_i^{\mathrm{CK},x} \leq \omega_{Ci+n+1}^{\mathrm{CK},y}$ for all $i \in \omega$. Consequently, we have $\omega_{i^2}^{\mathrm{CK}} = \omega_i^{\mathrm{CK},x} \leq \omega_{Ci+n+1}^{\mathrm{CK},y} = \omega_{Ci+n+1}^{\mathrm{CK}}$ for all $i \in \omega$, which implies $i^2 \leq Ci+n+1$ for all $i \in \omega$, which is obviously false (e.g. for $i \geq C+n+1$). So x is not reducible to an ITRM-c-random real. $\square$

Corollary 45. There are $0 <_{\mathrm{ITRM}} x, y < 0'_{\mathrm{ITRM}}$ such that $x \perp_{\mathrm{ITRM}} y$ and neither x nor y is ITRM-c-random.

Proof. Let $(a_n|n \in \omega)$ and $(b_n|n \in \omega)$ be two strictly increasing sequences of natural numbers such that, for every $C \in \omega$, there are $k, l \in \omega$ with $a_k > b_{Ck}$ and $b_l > a_{Cl}$. For example, we may take $a_1 = b_1 = 1$, and then recursively $a_{(2n+1)!} = b_{(2n+1)!} + (n-1)(2n+1)! + 1$, $b_{(2n)!} = a_{(2n)!} + (n-1)(2n)! + 1$ and $a_{k+1} = a_k + 1$, $b_{k+1} = b_k + 1$ if $k+1$ is not of the form $(2n+1)!$ or $(2n)!$, respectively. Now, by Theorem 43, we find reals x, y so that $\omega_i^{\mathrm{CK},x} = a_i$ and $\omega_i^{\mathrm{CK},y} = b_i$ for all $i \in \omega$. x, y are ITRM-computable from $0'_{\mathrm{ITRM}}$ and not ITRM-c-random by the same argument as in Theorem 44. To see that $x \perp_{\mathrm{ITRM}} y$, assume for a contradiction that one is reducible to the other, without loss of generality $x \leq_{\mathrm{ITRM}}^n y$. By Lemma 41, there is hence $C \in \omega$ such that $\omega_i^{\mathrm{CK},x} \leq \omega_{Ci+n+1}^{\mathrm{CK},y}$ for all $i \in \omega$. Pick $0 < k \in \omega$ such that $a_k > b_{(C+n+1)k}$. Then $\omega_{(C+n+1)k}^{\mathrm{CK},y} = \omega_{b_{(C+n+1)k}}^{\mathrm{CK}} < \omega_{a_k}^{\mathrm{CK}} = \omega_k^{\mathrm{CK},x} \leq \omega_{Ck+n+1}^{\mathrm{CK},y}$; consequently, we have $(C+n+1)k < Ck+n+1$, which is impossible for $k > 0$. $\square$

Remark: The same strategy allows the construction of arbitrarily long finite and in fact a countable set of pairwise incomparable, reals $<_{\mathrm{ITRM}} 0'_{\mathrm{ITRM}}$ which are neither ITRM-c-random nor even ITRM-reducible to a ITRM-c-random real.

5 Autoreducibility for Infinite Time Register Machines

Intuitively, a real x is called autoreducible if each of its bits can be effectively recovered from its position in x and all other bits of x. A discussion of the classical notion of autoreducibility can, for example, be found in [6]. We want to consider how this concept behaves in the context of infinitary machine models of computation. For the time being, we focus on ITRMs - but the notion of course makes sense for other types like the Infinite Time Turing

Machines (ITTMs, see [7]), Ordinal Turing machines and Ordinal Register Machines (OTMs and ORMs, see e.g. [15]) etc. as well.

Definition 46. For $x \in^{\omega} 2$, we define $x_{\backslash n}$ as x with its nth bit deleted (i.e. the bits up to n are the same, all further bits are shifted one place to the left). We say that x is ITRM-autoreducible if and only if there is an ITRM-program P such that $P^{x_{\backslash n}}(n) \downarrow = x(n)$ for all $n \in \omega$. x is called totally incompressible if and only if it is not ITRM-autoreducible, i.e. there is no ITRM-program P such that $P^{x_{\backslash n}}(n) \downarrow = x(n)$ for all $n \in \omega$. If there is such a program, then we say that P autoreduces x, P is an autoreduction for x or that x is autoreducible via P.

Corollary 47. No totally incompressible x is ITRM-computable or even recognizable. $0'_{\mathrm{ITRM}}$, the real coding the halting problem for ITRMs, is ITRM-autoreducible.

Proof. Clearly, if P computes x, then P is also an autoreduction for x. If x is recognizable and P recognizes x, we can easily retrieve a deleted bit by plugging in 0 and 1 and letting P run on both results to see for which one P stops with output 1. (The same idea works for finite subsets instead of single bits.) For $0'_{\mathrm{ITRM}}$, if a program index j is given, it is easy to determine some index $i \neq j$ corresponding to a program that works in exactly the same way (by e.g. adding a meaningless line somewhere), so that the remaining bits allow us to reconstruct the jth bit. $\square$

Remark: The autoreducibility of $0'_{\mathrm{ITRM}}$ also follows from the first part of the corollary and the recognizability of $0'_{\mathrm{ITRM}}$ (see [2]).

Definition 48. Let $x \in^{\omega} 2$, $i \in \omega$. Then $\mathrm{flip}(x, i)$ denotes the real obtained from x by just changing the ith bit, i.e. $x\Delta\{i\}$.

In the classical setting, no random real is autoreducible. This is still true for ITRM- and ITRM-c-randomness:

Theorem 49. If x is ITRM-random or ITRM-c-random, then x is totally incompressible.

Proof. Assume that x is autoreducible via P. We show that x is not ITRM-random. Let X be the set of all y which are autoreducible via P. Obviously, we have $x \in X$. X is certainly decidable: Given y, use a halting problem solver for P to see whether $P^{y_{\backslash n}}(n) \downarrow$ for all $n \in \omega$. If not, then $y \notin X$. Otherwise, carry out these ω many computations and check the results one after the other.

Since X is ITRM-decidable, it is provably Δ^1_2, which implies that X is measurable.

We show that X must be of measure 0. To see this, assume for a contradiction that $\mu(X) > 0$. Note first, that, whenever y is P-autoreducible and z is a real that deviates from y in exactly one digit (say, the ith bit), then z is not P-autoreducible (since P will compute the ith bit wrongly).

By the Lebesgue density theorem, there is an open basic interval I (i.e. consisting of all reals that start with a certain finite binary string s length $k \in \omega$) such that the relative measure of X in I is $> \frac{1}{2}$. Let $X' = X \cap I$, and let X'_0 and X'_1 be the subsets of X' consisting of those elements that have their $(k+1)$th digit equal to 0 or 1, respectively. Clearly, X'_0 and X'_1 are measurable, $X'_0 \cap X'_1 = \emptyset$ and $X' = X'_0 \cup X'_1$. Now define $\bar{X}'_0$ and $\bar{X}'_1$ by changing the $(k+1)$th bit of all elements of X'_0 and X'_1, respectively. Then all elements of $\bar{X}'_0$ and $\bar{X}'_1$ are elements of I (as we have not changed the first k bits), none of them is P-autoreducible (since they all deviate from P-autoreducible elements by exactly one bit, namely the kth), $\bar{X}'_0 \cap \bar{X}'_1 = \emptyset$ (elements of the former set have 1 as their $(k+1)$th digit, for elements of $\bar{X}'_1$ it is 0) and $\mu(\bar{X}'_0) = \mu(X'_0)$, $\mu(\bar{X}'_1) = \mu(X'_1)$ (as the $\bar{X}'_i$ are just translations of the X'_i). As no element of the $\bar{X}'_i$ is P-autoreducible, we have $(\bar{X}'_0 \cup \bar{X}'_1) \cap X' = \emptyset$. Let $\bar{X}' := \bar{X}'_0 \cup \bar{X}'_1$.

Then we have $\mu_I(\bar{X}') = \mu_I(\bar{X}'_0 \cup \bar{X}'_1) = \mu_I(\bar{X}'_0) + \mu_I(\bar{X}'_1) = \mu_I(X'_0) + \mu_I(X'_1) = \mu_I(X') > \frac{1}{2}$ (where μ_I denotes the relative measure for I). So X' and $\bar{X}'$ are two disjoint subsets of I both with relative measure $> \frac{1}{2}$, a contradiction.

For the analogous statement for ITRM-c-randomness, we proceed similarly, taking I to be an interval in which $X \cap I$ is comeager instead. That such an I exists can be seen as follows: Suppose that X is not meager. As above, X is ITRM-decidable, hence provably Δ^1_2 and therefore has the Baire property. Then, there is an open set U such that $(X \setminus U) \cup (U \setminus X)$ is meager. In particular, U is not empty. Hence X is comeager in U. As U is open, there is a nonempty open interval $I \subseteq U$. It is now obvious that $X \cap I$ is comeager in I, so I is as desired. We then use the same argument as above, noting that two comeager subsets of I cannot be disjoint. $\square$

Corollary 50. Let x be Cohen-generic over $L_{\omega_\omega^{\mathrm{CK}}}$. Then x is totally incompressible.

Proof. This follows immediately from Theorem 49 and Corollary 19. $\square$

Remark: When we demand that every finite subset of the bits of x (instead of every single bit) can be effectively obtained from the remaining bits and the positions of the missing bits by some ITRM-program, we get the notion

of **strong** autoreducibility. Strong autoreducibility is a strictly stronger notion than autoreducibility: If x is Cohen-generic over $L_{\omega_\omega^{\text{CK}}}$, then $y := x \oplus x$ is clearly autoreducible; however, by Corollary 50, y is not strongly autoreducible, as a procedure for obtaining even just the $2n$th and $2n+1$th bit of y from the remaining bits for every n would lead to an autoreduction for x.

Definition 51. Denote by IC_{ITRM} and RA_{ITRM} the set of totally incompressible and ITRM-random reals, respectively.

Corollary 52. The set of ITRM-autoreducible reals has measure 0 and is meager.

Proof. The proof of Theorem 49 shows that, for any ITRM-program P, the set of reals autoreducible via P has measure 0. As there are only countable many programs, the result follows. Meagerness follows from Corollary 50. □

By Theorem 49, we have $\text{RA}_{\text{ITRM}} \subseteq \text{IC}_{\text{ITRM}}$. By now, we have observed various similarities between totally incompressibility and randomness. However, the converse of Theorem 49 fails for ITRM-randomness:

Proposition 53. $\text{IC}_{\text{ITRM}} \nsubseteq \text{RA}_{\text{ITRM}}$, i.e. there is a real x such that x is totally incompressible, but not ITRM-random.

Proof. Let y be ITRM-c-random and let $x = y \oplus 0$. Then x is not ITRM-c-random since $[0,1] \oplus 0$ is ITRM-decidable and meager. Moreover, x is totally incompressible: For an autoreduction P for x would immediately lead to an autoreduction for y: To determine the nth bit of y given $y_{\setminus n}$, we only need to run $P^{(y \oplus 0)_{\setminus 2n}}$, where $(y \oplus 0)_{\setminus 2n}$ is just $y_{\setminus n} \oplus 0$ with an extra 0 inserted at the $2n$th place. However, by Theorem 49, y is totally incompressible. □

6 Conclusion and further work

The most pressing issue is certainly to strengthen the parallelism between ITRM-randomness and ML-randomness by studying the corresponding notion for sets of Lebesgue measure 0 rather than meager sets. Moreover, in focusing on decidable sets as the relevant tests for ITRM-c-randomness, we deviate from the spirit of ML-randomness. An ITRM-based notion closer to the idea of ML-randomness would be 'ITRM-ML-randomness': A set X is an ITRM-ML-test if and only if there is an ITRM-program P such that (1) $P(i)$ computes (in an appropriate coding) a set U_i of rational intervals with $\mu(\bigcup U_i) \leq 2^{-i}$ for all $i \in \omega$ and (2) $X = \bigcap_{i \in \omega} \bigcup U_i$. A real x is then called ITRM-ML-random if it is not an element of any ITRM-ML-test. Note that,

by solvability of the bounded halting problem for ITRMs, every ITRM-ML-test is ITRM-decidable, so that ITRM-ML-randomness is a weaker notion than ITRM-randomness. We presently do not know, however, whether it is strictly weaker.

Still, ITRM-c-randomness shows an interesting behaviour, partly analogous to ML-randomness, though by quite different arguments. Similar approaches are likely to work for other machine models of generalized computations, in particular ITTMs ([7]) (which were shown in [3] to obey the analogue of the non-meager part of Theorem 9) and ordinal Turing Machines ([14]) (for which the analogues of both parts of Theorem 9 turned out to be independent of ZFC) which we study in ongoing work. This further points towards a more general background theory of computation that allow unified arguments for all these various models as well as classical computability. Furthermore, we want to see whether the remarkable conceptual stability of ML-randomness (for example the equivalence with Chaitin randomness or unpredictabiliy in the sense of r.e. Martingales, see e.g. sections 6.1 and 6.3 of [6]) carries over to the new context.

7 Acknowledgements

I am indebted to Philipp Schlicht for several helpful discussions of the results and proofs and suggesting various crucial references. I also thank the referees of [4] for suggesting several simplifications and clarifications. Finally, I thank the anonymous referee of this paper for various improvements of the presentation.

References

[1] M. Carl. The distribution of ITRM-recognizable reals. In: S.B. Cooper, A. Dawar, M. Hyland and B. Löwe (eds.). Turing Centenary Conference: How the World Computes. Annals of Pure and Applied Logic Volume 165 (9), (2014), 1353-1532

[2] M. Carl. Optimal Results on ITRM-recognizability. To appear in the Journal of Symbolic Logic. Preprint available at arXiv:1306.5128v1 (2013)

[3] M. Carl, P. Schlicht. Infinite Computations with Random Oracles. To appear in the Notre Dame Journal of Formal Logic. Preprint available at arXiv:1307.0160 (2014)

[4] M. Carl. Algorithmic Randomness for Infinite Time Register Machines. In: A. Beckmann, E. Csuhaj-Varju and Klaus Meer (eds.) Language, Life and Limits. 10th conference on Computability in Europe. Lecture Notes in Computer Science 8493 (2014), 84-92

[5] N.J. Cutland. Computability - An introduction to recursive function theory. Cambridge University Press (1980)

[6] R.G. Downey, D. Hirschfeldt. Algorithmic Randomness and Complexity. Theory and Applications of Computability. Springer Berlin LLC (2010)

[7] J. Hamkins, A. Lewis. Infinite Time Turing Machines. Journal of Symbolic Logic 65(2), (2000), 567-604

[8] G. Hjorth, A. Nies. Randomness via effective descriptive set theory. Journal of the London Mathematical Society (2) 75 (2007), 495-508

[9] P. Koepke, R. Miller. An enhanced theory of infinite time register machines. In: Logic and Theory of Algorithms. A. Beckmann et al, eds., Lecture Notes in Computer Science 5028 (2008), 306-315

[10] M. Carl, T. Fischbach, P. Koepke, R. Miller, M. Nasfi, G. Weckbecker. The basic theory of infinite time register machines. Archive for Mathematical Logic 49:249-273 No. 4, (2010) 850-856

[11] R. Jensen, C. Karp. Primitive Recursive Set Functions. Axiomatic Set Theory, Proceedings of Symposia in Pure Mathematics, XIII, Part I, Providence, Rhode Island: American Mathematical Society, (1971), 143-176

[12] A. Kanamori. The higher infinite. Large Cardinals in Set Theory from Their Beginnings. Springer Monographs in Mathematics. Springer (2005)

[13] Y.N. Moschovakis. Descriptive Set Theory. Studies in Logic and the Foundations of Mathematics. vol. 100, North-Holland Publishing Company. New York (1980)

[14] P. Koepke. Turing computations on ordinals. Bulletin of Symbolic Logic 11 (2005), 377-397

[15] P. Koepke. Ordinal Computability. Mathematical Theory and Computational Practice. K. Ambos-Spies et al, eds., Lecture Notes in Computer Science 5635 (2009), 280-289

[16] A.R.D. Mathias. Provident sets and rudimentary set forcing. To appear in Fundamenta Mathematicae. Preprint available at `https://www.dpmms.cam.ac.uk/~ardm/fifofields3.pdf`

[17] G. Sacks. Higher recursion theory. Perspectives in Mathematical Logic, Volume 2. Springer Berlin (1990)

[18] S. Simpson. An extension of the recursively enumerable Turing degrees. Journal of the London Mathematical Society. Second Series, 75 (2007), 287-297,

[19] S. Simpson, G. Weitkamp. High and Low Kleene Degrees of Coanalytic Sets. Journal of Symbolic Logic, 48(2) (1983), 356-368.

[20] R.I. Soare. Recursively Enumerable Sets and Degrees: A study of computable functions and computably generated sets. Perspectives in Mathematical Logic. Springer Heidelberg (1987)